CNN Auto-Encoder Network Using Dilated Inception for Image Steganography

Ismail Kich, El Bachir Ameer, and Youssef Taouil

Computer Science Research Laboratory, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco

Abstract

Numerous studies have used convolutional neural networks (CNNs) in the field of information concealment as well as steganalysis, achieving promising results in terms of capacity and invisibility. In this study, we propose a CNN-based steganographic model to hide a color image within another color image. The proposed model consists of two sub-networks: the hiding network is used by the sender to conceal the secret image; and the reveal network is used by the recipient to extract the secret image from the stego image. The architecture of the concealment sub-network is inspired by the U-Net auto-encoder and benefits from the advantages of the dilated convolution. The reveal sub-network is inspired by the auto-encoder architecture. To ensure the integrity of the hidden secret image, the model is trained end to end: rather than training separately, the two sub-networks are trained simultaneously a pair of networks. The loss function is elaborated in such a way that it favors the quality of the stego image over the secret image as the stego image is the one that comes under steganalysis attacks. To validate the proposed model, we carried out several tests on a range of challenging publicly available image datasets such as ImageNet, Labeled Faces in the Wild (LFW), and PASCAL-VOC12. Our results show that the proposed method can dissimulate an image into another one with the same size, reaching an embedding capacity of 24 bit per pixel without generating visual or structural artefacts on the host image. In addition, the proposed model is generic, that is, it does not depend on the image's size or the database source.

Keywords: Information security, Image steganography, Dilated convolution, CNN, Auto-encoder

1. Introduction

New communication technologies have made the exchange of information easier and cheaper. However, these technologies come with their challenges: there is an increased risk these days of sensitive information being disclosed and modified by hackers and cyber-attackers on the Internet. Therefore, given this prevalent electronic warfare on open networks, securing and protecting the information exchanged has become a crucial priority. Organizations and individuals are therefore resorting to relying on techniques such as encryption and information concealment algorithms to overcome security problems [1].

Steganography is the art of invisibly concealing information into innocent-looking digital files so that they can be transferred without anyone noticing them. Unlike cryptography, which encrypts transmitted secure data, steganography simply hides the data from a third party. Images are the most widely used medium in steganography; an image is easy to handle; and

Received: Apr. 23, 2021
 Revised : Jun. 22, 2021
 Accepted: Aug. 31, 2021

Correspondence to: Ismail Kich
 (kichsma@gmail.com)
 ©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

considering images are pretty common on the Internet, they do not arise suspicion [2]. Concealing information into a carrier (cover) image produces a new image (stego image); this latter image carries the information that is secretly transferred without the awareness of a third-party. Image steganography finds its application in many areas such as the transmission of confidential data in the military or industrial field and the exchange of encryption keys between users [3].

The performance of a steganographic algorithm is generally measured based on three criteria: imperceptibility, capacity, and security. Imperceptibility allows us to measure the similarity between the stego image and the original cover image. Capacity refers to the average number of bits inserted in each pixel of the cover image; it is measured in bit per pixel (bpp). Security expresses the possibility of secret information being detected by third-party steganalysis attacks [4]. Although highly effective, these three criteria are in conflict: improving one affects the other two. When a larger volume of secret data is concealed, capacity becomes larger; however, security may worsen, and imperceptibility risks becoming noticeable. It is, therefore, essential to seek a compromise between the values of these parameters, and especially one that makes it possible to obtain good capacity while retaining acceptable values of the other parameters [5].

Traditional image steganography schemes can hide a payload capacity of about 0.4 bpp without the stego image being detected. With a small payload, the message can be safely hidden. When the payload reaches about 4 bpp, the imperceptibility to the human eye is challenged, and the probability of the stego image being detected by steganalysis becomes high [6]. To achieve a good compromise between a high capacity and a good imperceptibility of the stego image, a new generic image steganographic scheme is proposed in this study. We hide a color image (secret information) in a color image (cover) of the same size. Motivated by the remarkable results of using a convolutional neural network (CNN) in steganography as well as in steganalysis, following are the contributions of this study: first, the proposed steganographic scheme is an end-to-end trained deep learning model; we used dilated inception convolution to extend the receptive field without down-sampling the size of the features map. Second, as in the U-Net structure, the feature maps of the deep blocks are cascaded with the feature maps of the previous blocks, so that the network continues to learn about these features throughout the learning network. Third, a new loss function is used during training, so that the generated stego image, which is controlled by steganalysts, can preserve high fi-

delity. Our previous work [7] was built with a basic convolution layers and trained with loss function based on weighted L_2 loss. The test results suggest that the proposed model outperforms our previous model and existing models.

The rest of the paper is organized as follows: Section 2 describes some recent steganography-related works based on deep learning. Our proposed steganographic scheme is described in Section 3. Experimental results of our neural network are presented and discussed in Section 4. In Section 5, we present our conclusions.

2. Related Works

For practical reasons, most traditional image steganography algorithms use the spatial least significant bit (LSB) and their improved extension, to embed secret information into the cover image. The secret information bits are hidden directly in the lowest bit of each pixel by simple substitution, which can hide large capacity with good imperceptibility [8]. However, these methods are vulnerable to statistical attacks and can be easily detected [9]. Subsequently, many safer steganography methods have been proposed with the aim of maximizing integration distortion while attempting to improve the ability to integrate. These methods adopt several techniques for evaluating and selecting noisy or complex texture regions for incorporating secret information. The concealment in this case is adapted to the content or texture of the cover image. The following methods, HUGO [6], WOW [10], and S-UNIWARD [11], are robust to steganalysis attacks, but they are highly dependent on the content of the cover image, making it difficult to calculate the average payload capacity [12].

In a steganography scheme, a secret text message requires a perfect reconstruction (without a single error) at the recipient end. This condition can be alleviated when the secret message is an image because it can be reconstructed by the receiver with an acceptable error rate without losing the information integrity of the hidden image. This gives us the possibility to adopt lossy steganographic algorithms taking an image as a secret message. Researchers in the field of steganography have recently been able to hide one image within another using deep learning; furthermore, they have already made significant progress in terms of payload capacity and imperceptibility against different steganalysis attacks. In [13–16], the authors propose hiding the image into another image of the same size, using a different architecture based on a CNN. The location where to hide information bits is selected by ingenious networks of neurons.

Figure 1. Image steganography architecture based on deep neural network.

In [13], the author proposed a deep learning steganography-based generic encoder-decoder network. The model is composed of three sub-nets: prep-network, hiding network, and reveal network, as shown in Figure 1; each sub-network uses a sequence of 5 convolution layers that has 50 filters each of $\{3 \times 3, 4 \times 4, 5 \times 5\}$ patches. The network is trained end to end to ensure the integrity of the concealment and extraction process. It can hide a color image in another color image of the same size, that is, a capacity of 24 bpp. However, the architecture used in this model is considerably complicated and the color of generated stego images is distorted. In [14], the authors complete the same task, except that they used gray-scale image as the secret image. The proposed architecture is composed only of two networks (hiding and reveal). Each network uses a sequence of 3×3 convolutions and ReLU activation function in each layer, except the last layer, which uses 1×1 convolution without the ReLU activation function. This model also offers a large hiding capacity (8 bpp) with good imperceptibility. Nevertheless, the stego image still suffers from color distortion. In [15], the authors proposed an auto-encoder like U-Net structured with a CNN. In the hiding phase, it uses a sequence of 4×4 convolution layers followed by a leaky ReLU activation function and batch normalization (BN) operation, except the output layer that uses the Sigmoid activation function to calculate the stego image. This method has significant advantages in terms of capacity (24 bpp) and imperceptibility. However, the architecture of this model is not generic on the image's inputs size. The authors in [16] proposed to hide a gray-scale image into Y channel of the cover color image; the model called an invisible steganography via generative adversarial network (ISGAN) uses the inception module in the

hiding network. To improve the security of the stego image, the generative adversarial networks is introduced to minimize the divergence between the empirical probability distributions of stego images and natural images.

In summary, although these studies produced good concealment results, they still present limitations, either at the level of the hidden information capacity, or at the level of the color distortion of the stego images, or that the proposed models work well only on some specified image size. In the present study, we managed to realize a generic model for different sizes of color images, with a large hidden capacity (24 bpp) and a good imperceptibility of the stego and extracted images. The proposed model takes advantage of the features of U-Net network and dilated inception network.

3. Our Approach

Inspired by Baluja's work [13] and U-Net structure [17], our proposed model uses only two networks: the hiding network and the reveal network, Figure 2. The prep-network in [13] extracts the features of the secret image before the dissimulation phase, after which these features are inputs to the hiding network. In our model, the extraction of the secret image's features is done simultaneously with the dissimulation operation in the hiding network; the model learns to extract the features and hide them simultaneously. It is trained end-to-end: The model is not first trained on hiding and then on revealing separately; instead, it is trained to hide the secret image and reveal it simultaneously. In this study, the hiding network is inspired by the U-Net auto-encoder architecture in which we propose to use dilated convolution; it creates a stego image from the cover image and

Figure 2. Architecture of proposed scheme.

the secret image. The CNN layers are used to learn the hierarchy of the image features, a hierarchy ranging from low-level to high-level specific features. Thus, the hiding network learns the features of both secret and cover images, which enables it to hide the features of the secret image in the features of the cover image. In other words, the objective is to compress and distribute the bits of secret image on all the bits available on the cover image. The reveal network extracts the secret image from the stego image using fully classical convolutional network.

3.1 Dilated Convolution

In deep learning, convolution on image is used to extract features. However, it is usually followed by pooling operations to keep only the high-level features. This can degrade the resolution of the picture, but it is beneficial in high-level computer vision tasks such as classification problems. In this study, reducing the resolution can be problematic, because the pooling operation is irreversible and can cause the loss of spatial information, which may prevent the reconstruction of the information of the small objects in the image. To resolve this problem, we use the dilated convolution, which has been referred to in the past as “convolution with a dilated filter.” Given a filter f and a discrete signal g , the classic convolution of f and g produces the signal h calculated as follows:

$$h(t) = \sum_k f(k) g(t - k).$$

The dilated convolution of f and g by rate r is the following

Figure 3. The dilated convolution with rates of 1, 2 and 4.

signal h_{dil} :

$$h_{dil}(t) = \sum_k f(k) g(t - rk), \quad (1)$$

while r represents the dilation rate. Figure 3 gives an example of the dilation convolution for images. In the left image, $r = 1$ which is exactly the classic convolution. In the images in the middle and right, $r = 2$ and $r = 4$ respectively, we skip 2 and 4 values in the convolution.

The dilated convolution operator can apply the same filter (as in classic convolution) at different ranges using different dilation rates r . The dilation convolution can support the expansion of the receptive field without loss of the image’s resolution, which is what needed in our topic.

3.2 Hiding Network Architecture

The hiding network receives as input the cover image and the secret image; both are concatenated into a 6-channel tensor. It is composed of two types of layers, one with classical convolution and others with a dilated convolution module. This latter is inspired by what is called “the inception module,” GoogLeNet

Figure 4. The dilated inception module.

teams developed it in the ILSVR14 competition. The module aims to extract the multi-scale contextual information [18]. Several other ameliorated versions were proposed later. The key idea of this module is to apply convolutions with kernels of different sizes, which helps to extract multi-scale features in receptive fields of different sizes. As explained in the previous section, to extract features of both images without degrading the resolution, we propose the dilated inception module. As schemed in Figure 4, this module is composed of 4 branches, the 1st one is 1×1 convolution followed by ReLU activation. The 2nd, 3rd and 4th branches begin with 1×1 convolution and ReLU activation, after which dilated 3×3 convolution follows with rates of 1, 2, 3, respectively, to extract features on a neighbor of different sizes. Finally, the four obtained tensors are concatenated to produce the input of the next layer.

The BN is introduced into the hiding network to speed up the training procedure. The details of the hiding structure are described in Figure 5 and Table 1. The encoder and decoder networks both use a sequence of 3×3 convolutions with ReLU activation and a dilated inception module. Like in U-Net structure, each layer in the encoder is cascaded with the feature map of the corresponding layer in the decoder; the goal is to make sure that the network also learns the features map of the previous layers. At the last layer, the 3×3 convolutions are applied to compress the convoluted feature channels into a 3-channel characteristic map, followed by a BN operation and a Sigmoid activation to construct the stego image.

Figure 5. Hiding network scheme.

Table 1. Architecture details of the hiding network

Layers	Process	Output layers size
Inputs	Concatenate	$N \times N \times 6$
Layer 1	3×3 Conv+BN+ReLU	$N \times N \times 64$
Layer 2	Dilated Inception Block	$N \times N \times 128$
Layer 3	Dilated Inception Block	$N \times N \times 256$
Layer 4	Dilated Inception Block	$N \times N \times 512$
Layer 5	Dilated Inception Block	$N \times N \times 512$
Layer 6	Dilated Inception Block	$N \times N \times 512$
Layer 7	Dilated Inception Block	$N \times N \times 256$
Layer 8	Dilated Inception Block	$N \times N \times 128$
Layer 9	3×3 Conv+BN+ReLU	$N \times N \times 64$
Output	3×3 Conv+BN+Sigmoid	$N \times N \times 3$

3.3 Reveal Network Architecture

The input in the reveal network is the stego image received from the hiding network. As shown in Figure 6 and Table 2, to each layer, we apply 3×3 classical convolutions, followed by a BN operation and a ReLU activation function. For the last layer, the activation function is a Sigmoid; the obtained matrix then reveals the secret image.

3.4 Loss Function

In image steganography, as we dissimulate a secret image into a cover image, the loss function contains two errors: error of the cover-stego images and the error of the secret-revealed images. A question arises: Do the two errors have the same importance in the context of steganography? The stego image is the one transmitted in the communication channel; therefore, it is the

Table 2. Architecture details of the reveal network

Layers	Process	Output layers size
Inputs	$3 \times 3 \text{Conv} + \text{BN} + \text{ReLU}$	$N \times N \times 6$
Layer 1	$3 \times 3 \text{Conv} + \text{BN} + \text{ReLU}$	$N \times N \times 64$
Layer 2	$3 \times 3 \text{Conv} + \text{BN} + \text{ReLU}$	$N \times N \times 128$
Layer 3	$3 \times 3 \text{Conv} + \text{BN} + \text{ReLU}$	$N \times N \times 256$
Layer 4	$3 \times 3 \text{Conv} + \text{BN} + \text{ReLU}$	$N \times N \times 128$
Layer 5	$3 \times 3 \text{Conv} + \text{BN} + \text{ReLU}$	$N \times N \times 64$
Output	$3 \times 3 \text{Conv} + \text{BN} + \text{Sigmoid}$	$N \times N \times 3$

one facing the danger of steganalysis attacks; thus, the model's invisibility depends on the quality of the stego image. Hence, it is mandatory that it does not arise suspicion. As for the secret image, in general, it still can keep its usefulness to the receiver even if it does not conserve the same quality as before it was dissimulated. Therefore, the error of the cover-stego images has more priority than the error of the secret-revealed images.

In the proposed model, end-to-end trained network is used to first produce the stego image from cover and secret images, and second to reconstruct the revealed image from the stego image. While learning, this neural network model needs to estimate millions of parameters; the estimation is achieved by minimizing the weighted loss function, which, as mentioned before, contains two errors. Let C and S represent, respectively, the cover and secret images, and let C' and S' represent, respectively, the stego image and the revealed secret image. The error between C and C' is measured by the mean absolute error L_1 , plus the variance loss. To prioritize the error between C and C' over the error between S and S' , the variance loss is added because the encoding and decoding of images generates modifications on the reconstructed images, so the use of variance loss in the encoding network gives prevents the concentration of these modifications on some areas and indicates to the neural network that the loss must be distributed over the entire im-

age. This variance loss is calculated across the image's height, width, and channel. The error between S and S' is calculated by the mean squared error L_2 . Therefore, The network is trained end-to-end to optimize the following loss function, $Loss$:

$$Loss = \frac{1}{2} (L_1(C, C') + Var(L_1(C, C'))) + L_2(S, S'), \quad (2)$$

where

$$L_1(C, C') = \frac{1}{n} \sum_{i=1}^n |C_i - C'_i|, \quad (3)$$

$$L_2(S, S') = \frac{1}{n} \sum_{i=1}^n (S_i - S'_i)^2, \quad (4)$$

and n is the number of pixels in each image.

4. Experiment Result

In this part, we present and discuss the results of our experiments. Image datasets such as ImageNet [19], Labeled Faces in the Wild (LFW) [20], and PASCAL-VOC12 [21] have been set up to test our training network system. Each database is randomly divided into three datasets, namely: training, validation, and test. All training results have been validated by the validation set. Results reported in this document are performed on the test sets. The Adam optimizer was used with an initial learning rate of 0.001, which was then reduced to 0.0001 after 150 epochs of training. All weights of our model were initialized randomly using the Xavier initialization [22].

We use the peak signal-to-noise ratio (PSNR) [23] and the structural similarity index measure (SSIM) [24] as metrics to evaluate our proposed model's performance. The PSNR confirms the imperceptibility by calculating the error between corresponding pixels. The more the cover and stego image are close, the higher is the value of PSNR. It is calculated as follows:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{L_2(C, C')} \right). \quad (5)$$

The SSIM estimates the similarity of two images by calculating three terms namely: luminance, contrast and structure. The closer the cover image and the stego image, the closer the value of SSIM is to 1. The SSIM between two images can be calculated as follows:

$$l(x, y) = \frac{2\mu_x\mu_y + \tau_1}{\mu_x^2 + \mu_y^2 + \tau_1}, \quad (6)$$

Figure 6. Reveal network scheme.

Table 3. PSNR and SSIM values for different models

Model	Size	Cover-stego		Secret-revealed	
		PSNR	SSIM	PSNR	SSIM
Our previous model [7]	32×32×3	35.25	0.9845	32.53	0.9625
	64×64×3	35.35	0.9710	32.70	0.9535
	128×128×3	36.00	0.9692	33.33	0.9445
	256×256×3	35.22	0.9554	32.21	0.9338
Proposed $L_2 + L_2$	32×32×3	35.50	0.9893	32.98	0.9689
	64×64×3	36.41	0.9770	33.53	0.9509
	128×128×3	36.15	0.9751	33.40	0.9501
	256×256×3	35.97	0.9686	33.01	0.9479
Proposed $L_1 + V + L_2$	32×32×3	36.50	0.9860	30.01	0.9190
	64×64×3	37.36	0.9820	30.10	0.9109
	128×128×3	37.71	0.9793	31.62	0.9090
	256×256×3	37.83	0.9786	31.77	0.9077

$$c(x, y) = \frac{2\sigma_x\sigma_y + \tau_2}{\sigma_x^2 + \sigma_y^2 + \tau_2}, \quad (7)$$

$$s(x, y) = \frac{\sigma_{xy} + \tau_3}{\sigma_x\sigma_y + \tau_3}, \quad (8)$$

$$SSIM = l(x, y) c(x, y) s(x, y), \quad (9)$$

where x represent an original image and y represent its reconstructed image; μ_x and μ_y are respectively the mean values of the images x and y ; σ_x^2 and σ_y^2 are respectively the variances of x and y ; σ_{xy} is the co-variance of x and y . As for τ_1 , τ_2 and τ_3 , they are small positives constants to stabilize the division weak denominator.

For the first experiments, we trained our model on images with different sizes; the images were randomly selected and then resized from the ImageNet dataset. The results were then compared with our previous method cited in [7]. To accomplish the comparison, we first test the proposed method using the same loss function used in the previous model [7]; we call it "Proposed $L_2 + L_2$ ". Then, we use the loss function proposed in this article (see Section 3.4); we call it "Proposed $L_1 + V + L_2$ ". The values in the table are the average PSNR and SSIM for all the images of the test set. The PSNR in the cover-stego column measures the invisibility of the model, while the PSNR in the hidden-extracted column measures the visual similarity between the hidden and revealed secret image, they evaluate the integrity of the model.

As shown in Table 3, we were able to hide one image into another one with the same size, which allowed us to reach a 100%

Table 4. PSNR and SSIM values of our ImageNet trained algorithm on images from LFW and PASCAL-VOC12 datasets

Model	Images dataset	Cover-stego		Secret-revealed	
		PSNR	SSIM	PSNR	SSIM
Rehman et al. [14]	LFW	33.7	0.95	39.9	0.96
	P.-V.12	33.7	0.96	35.9	0.95
Zhang et al. [16]	LFW	34.63	0.9573	33.63	0.9429
	P.-V.12	34.49	0.9661	33.31	0.9467
Proposed	LFW	38.35	0.9659	34.90	0.9505
$L_2 + L_2$	P.-V.12	35.94	0.9731	32.95	0.9532
Proposed	LFW	40.03	0.9797	33.13	0.9280
$L_1 + V + L_2$	P.-V.12	37.40	0.9790	30.80	0.9094

payload (24 bpp) in the cover image with acceptable values of PSNR and SSIM. When we used the same loss function as in [7], the PSNR and the SSIM were better than those obtained in the previous work, regardless of whether it was used for the stego or for the secret image. The obtained PSNR was found to be better than the previous work for the stego image. However, for the secret-revealed image, the PSNR and the SSIM decreased slightly in comparison with the previous work; this is due to the fact that the proposed loss function favors the error cover-stego over the error secret-revealed. In summary, depending on the application's requirements, one can choose which error one wants to favor. It can be noted that for the proposed work with $L_1 + V + L_2$ loss function, the PSNR and SSIM values become better as the size of the cover image becomes bigger; when we use images with bigger size, there are more features that the model can be trained on; this explains why the results get better.

To test the portability of our steganographic system on other images from other images datasets, we used our already-trained model on ImageNet dataset and ran it on samples of images from PASCAL-VOC12 and LFW. We randomly selected 5,000 images from each dataset, and then we resized them to (300 × 300 × 3) to form pairs of secret-cover images of the test sets. Table 4 shows the results of comparison with models proposed in [14], [16] and our previous model [7]. We note that the models presented in [14] and [16] use only gray images as secret images, i.e., a masking capacity around 33.33% (8 bpp).

Based on the results shown in Table 4, we can say that the proposed model is extremely generic. Even if it is trained on ImageNet dataset, it can hide and recover images while preserving their imperceptibility regardless of the source of

these images. The imperceptibility values of the proposed method, which are shown in the column cover-stego, are higher than those of Rehman's [14] and Zhang's [16], whether with the old or the new loss function. However, for the secret image, Rehman's values are rivaling with the PSNR values when we use the old loss function. Rehman's values are better than the PSNR of the proposed model. But, for these PSNR values, the capacity of the proposed model is 3 times larger than Rehman's and Zhang's models.

For a qualitative control of our steganographic system. Figures 7 and 8 show four examples of images (covers and secrets) on ImageNet as well as the residual images, magnified 5 to 10 times, between the input images and their reconstructed images. The residual images are compared to those of our previous method [7].

The pixels of the residual image R are calculated as follows:

$$\forall 1 \leq i \leq n, R_i = \frac{|C_i - C'_i|}{M}, \quad (10)$$

where $M = \max_{1 \leq i \leq n} |C_i - C'_i|$, and n is the number of the image's pixels.

Results show that the stego and revealed images are visually identical to the original cover and secret images. The residual

image between the cover and stego images has the same shape or contours of the cover image despite hiding an image of the same size. The error between the secret image and the revealed image is a little larger than the error between the cover and stego image; the textures of the cover image appear faintly on the residual image between the secret image and the revealed image. However, on the revealed image, no remarkable distortion decreasing its quality can be observed. On the residual image of the previous model, unlike the proposed model, we can faintly remark some contours of the secret image. The variance loss added in the loss function helps in the distribution of the dissimulation error in the entire image instead of having it eventually concentrated in some areas, thus leading to better visual similarity between the cover and stego images. This high visual similarity of proposed method prevents the image being detected by steganalysts, which enhances the security of steganographic communication.

5. Conclusions and Perspectives

We proposed a new deep learning-based steganographic model that conceals a color image into another one of the same size with performances superior to traditional methods. The ba-

Figure 7. The residual images between the cover images sampled from test images from ImageNet and their corresponding stego images, enhanced 5 and 10 times.

Figure 8. The residual images between the secret images sampled from test images from ImageNet and their corresponding revealed images, enhanced 5 and 10 times.

sic module used in the hiding network is the dilated inception module to enlarge the receptive area within each features map without reducing the tensor length or width. In addition, the structure of the U-Net architecture is integrated so that during the decoding phase, the hiding network learns the features map of the corresponding layers in the encoding phase. During training, the proposed loss function combines L_1 -Loss and variance as error between the cover and stego images, and the L_2 -Loss between the secret and revealed images. Furthermore, we accomplished several tests using different analysis methods applied in the field of image steganography; results demonstrated the security of the proposed method and the good visual quality of both the stego and secret images even though the capacity reaches 24 bpp. The proposed loss function presents a good residual cover-stego image, although it does so at the expense of the visual quality of the revealed image. Therefore, the adopted loss function may have to be ameliorated. In addition, although increasing the image's size enriches the learning of the model, thus leading to better results, when the color image's size exceeds 300, it is difficult to run the training because of power computing issues. To address this issue, we reduced the batch to 8 for 256×256 images and 4 for 300×300 images. In the future, we intend to explore some possibilities that neu-

ral networks can provide to resolve this issue. We will also try to combine the concealment process with the generative adversarial networks to increase security while minimizing the divergence between stego and cover images.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

References

- [1] I. Hussain, J. Zeng, X. Qin, and S. Tan, "A survey on deep convolutional neural networks for image steganography and steganalysis," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 3, pp. 1228-1248, 2020. <https://doi.org/10.3837/tiis.2020.03.017>
- [2] Y. Taouil, E. B. Ameur, and M. T. Belghiti, "New image steganography method based on Haar discrete wavelet transform," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*. Cham, Switzerland: Springer, 2017, pp. 287-297. https://doi.org/10.1007/978-3-319-46568-5_30

- [3] B. Li, J. He, J. Huang, and Y. Q. Shi, "A survey on image steganography and steganalysis," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 2, pp. 142-172, 2011.
- [4] M. S. Subhedar and V. H. Mankar, "Current status and key issues in image steganography: a survey," *Computer Science Review*, vol. 13, pp. 95-113, 2014. <https://doi.org/10.1016/j.cosrev.2014.09.001>
- [5] S. Kumar, A. Singh, and M. Kumar, "Information hiding with adaptive steganography based on novel fuzzy edge identification," *Defence Technology*, vol. 15, no. 2, pp. 162-169, 2019. <https://doi.org/10.1016/j.dt.2018.08.003>
- [6] T. Pevny, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Information Hiding*. Heidelberg, Germany: Springer, 2010, pp. 161-177. https://doi.org/10.1007/978-3-642-16435-4_13
- [7] I. Kich, Y. T. El Bachir Ameer, and A. Benhfid, "Image steganography by deep CNN auto-encoder networks," *International Journal*, vol. 9, no. 4, pp. 4707-4716, 2020. <https://doi.org/10.30534/ijatcse/2020/75942020>
- [8] I. Kich, E. B. Ameer, and Y. Taouil, "Image steganography based on edge detection algorithm," in *Proceedings of 2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, Kenitra, Morocco, 2018, pp. 1-4. <https://doi.org/10.1109/ICECOCS.2018.8610603>
- [9] R. Meng, Q. Cui, and C. Yuan, "A survey of image information hiding algorithms based on deep learning," *Computer Modeling in Engineering & Sciences*, vol. 117, no. 3, pp. 425-454, 2018. <https://doi.org/10.31614/cmes.2018.04765>
- [10] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proceedings of 2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, Costa Adeje, Spain, 2012, pp. 234-239. <https://doi.org/10.1109/WIFS.2012.6412655>
- [11] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, article no. 1, 2014. <https://doi.org/10.1186/1687-417X-2014-1>
- [12] P. Wu, Y. Yang, and X. Li, "Image-into-image steganography using deep convolutional network," in *Advances in Multimedia information Processing – PCM 2018*. Cham, Switzerland: Springer, 2018, pp. 792-802. https://doi.org/10.1007/978-3-030-00767-6_73
- [13] S. Baluja, "Hiding images in plain sight: deep steganography," *Advances in Neural Information Processing Systems*, vol. 30, pp. 2069-2079, 2017.
- [14] A. ur Rehman, R. Rahim, S. Nadeem, and S. ul Hussain, "End-to-end trained CNN encoder-decoder networks for image steganography," in *Computer Vision - ECCV 2018 Workshop*. Cham, Switzerland: Springer, 2018, pp. 723-729. https://doi.org/10.1007/978-3-030-11018-5_64
- [15] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, "Reversible image steganography scheme based on a U-Net structure," *IEEE Access*, vol. 7, pp. 9314-9323, 2019. <https://doi.org/10.1109/ACCESS.2019.2891247>
- [16] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8559-8575, 2019. <https://doi.org/10.1007/s11042-018-6951-z>
- [17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham, Switzerland: Springer, 2015, pp. 234-241. https://doi.org/10.1007/978-3-319-24574-4_28
- [18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, 2015, pp. 1-9.
- [19] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "ImageNet: a large-scale hierarchical image database," in *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 248-255. <https://doi.org/10.1109/CVPR.2009.5206848>
- [20] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," in *Proceedings of International Workshop on Faces in Real-Life Im-*

ages: *Detection, Alignment, and Recognition*, Marseille, France, 2008.

- [21] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303-338, 2010. <https://doi.org/10.1007/s11263-009-0275-4>
- [22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy, 2010, pp. 249-256.
- [23] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings of 2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, 2010, pp. 2366-2369. <https://doi.org/10.1109/ICPR.2010.579>
- [24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004. <https://doi.org/10.1109/TIP.2003.819861>

Ismail Kich is a Ph.D. candidate at Computer science research laboratory, University Ibn Tofail, Morocco. His researchers are focused on image and signal processing, machine learning, deep learning, steganography and data hiding.

E-mail: kichsma@gmail.com

El Bachir Ameur is a Researcher Professor of Computer Sciences at the University of Ibn Tofail, Faculty of Science, Kenitra, Morocco. In 2002, he received his Ph.D. in Numerical Analysis and Computer Sciences from the University of Mohamed I Oujda, Morocco. His Ph.D. concerned approximation and reconstruction of 2D/3D data by spline and wavelet functions. His research interests include approximation and reconstruction of 2D/3D surfaces by spline and wavelets, signal and image processing, data hiding, machine learning, and deep learning. E-mail: ameurelbachir@yahoo.fr

Youssef Taouil obtained his Ph.D. degree from the Faculty of Sciences, Ibn Tofail University in 2018. He obtained his Engineering degree in Electronics and Embedded Systems from the National School of Applied Sciences at the same University in 2014. His researches are focused on image and signal processing, multi-resolution analysis and wavelets, steganography, data hiding, machine learning, and deep learning.

E-mail: taouilysf@gmail.com